

## MP#2: Report

### myheader.h:

There I declare packet structure as given in the project description.

### Client.c:

After making all connections, there I declare two pthreads: 'thread1' – for sending packets to server, 'thread2' – for receiving packets from server. Accordingly, on each pthreads I run two functions: 'send' and 'receive', respectively. Additionally, I created 'args' struct to pass arguments to these functions.

In 'send' function I generate given 'npkts' of random packets and arrival times according to the given mean of packet size and arrival time. For this I used created function 'exp\_dis' which calculates random according to the given mean of exponentially distributed function. At the end, I create one more packet with 'seqno=-1' to notify server that all packets were sent and received.

In 'receive' function I receive all packets from server, and calculate delay time. When I get packet with 'seqno' equal to 'npkts-1', which means it is the last packet, I stop receiving.

### Server.c:

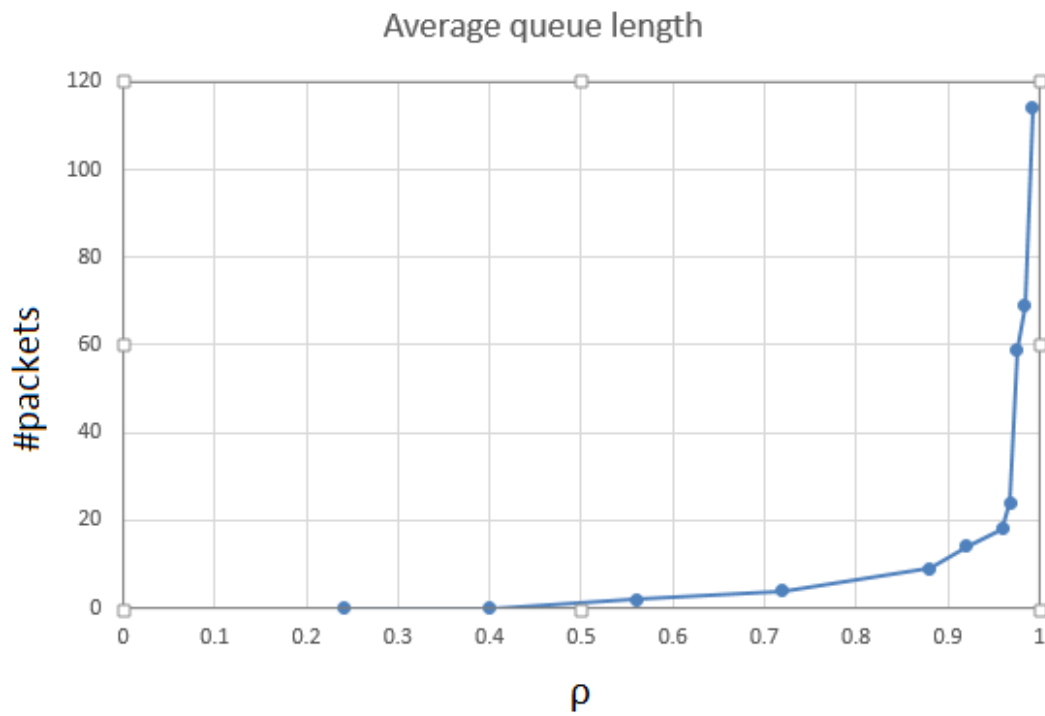
I created FIFO queue structure with all needed functions and variables.

After creating socket and starting server, start to receive packets. There is a child pthread function 'service' to work on packets and send it back to the client. 'Service' is called only when queue is not empty and other pthread is not working with queue already.

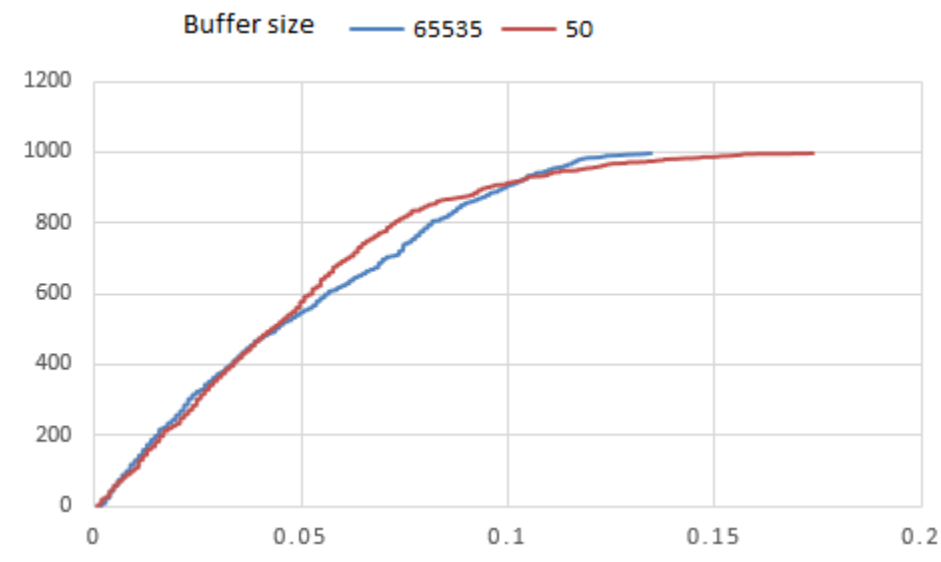
In 'service', proceed all packets in queue while queue is not empty. After getting a packet, calculate a service time and simulate delay. Then send this packet back to client, now with updated value of 'serverQueueLen'. Whenever queue becomes empty, thread stops its work.

```
[cs20132027@uni06 MM1]$ ./server 25000 65535
1444824497.442292000 0 + pkt 0
1444824497.443210000 0 + pkt 1
1444824497.472245000 2 - pkt 0 size 3588 serv time 0.028704000
1444824497.481249000 0 + pkt 2
1444824497.486246000 0 + pkt 3
1444824497.489239000 3 - pkt 1 size 1967 serv time 0.015736000
1444824497.501245000 0 + pkt 4
1444824497.505245000 0 + pkt 5
1444824497.514247000 4 - pkt 2 size 2922 serv time 0.023376000
1444824497.518237000 3 - pkt 3 size 328 serv time 0.002624000
1444824497.521247000 0 + pkt 6
1444824497.521256000 3 - pkt 4 size 205 serv time 0.001640000
1444824497.522256000 0 + pkt 7
1444824497.534246000 3 - pkt 5 size 1510 serv time 0.012080000
1444824497.536249000 2 - pkt 6 size 105 serv time 0.000840000
1444824497.537236000 1 - pkt 7 size 26 serv time 0.000208000
1444824497.542254000 0 + pkt 8
1444824497.548260000 0 + pkt 9
1444824497.563235000 2 - pkt 8 size 2521 serv time 0.020168000
1444824497.579234000 1 - pkt 9 size 1830 serv time 0.014640000
[cs20132027@uni06 MM1]$

[cs20132027@uni07 MM1]$ ./client 10.0.7.241 25000 120 10
1444824497.426031000 TX 0 size 3588
1444824497.427015000 TX 1 size 1967
1444824497.456181000 RX 0 size 3588 qlen 0 delay 0.030150000
1444824497.465003000 TX 2 size 2922
1444824497.470015000 TX 3 size 328
1444824497.473186000 RX 1 size 1967 qlen 1 delay 0.046171000
1444824497.485006000 TX 4 size 205
1444824497.489022000 TX 5 size 1510
1444824497.498180000 RX 2 size 2922 qlen 1 delay 0.033177000
1444824497.502170000 RX 3 size 328 qlen 2 delay 0.032155000
1444824497.505013000 TX 6 size 105
1444824497.505190000 RX 4 size 205 qlen 2 delay 0.020184000
1444824497.505994000 TX 7 size 26
1444824497.518178000 RX 5 size 1510 qlen 3 delay 0.029156000
1444824497.520177000 RX 6 size 105 qlen 2 delay 0.015164000
1444824497.521166000 RX 7 size 26 qlen 2 delay 0.015172000
1444824497.526012000 TX 8 size 2521
1444824497.532011000 TX 9 size 1830
1444824497.547192000 RX 8 size 2521 qlen 0 delay 0.021180000
1444824497.563179000 RX 9 size 1830 qlen 1 delay 0.031168000
[cs20132027@uni07 MM1]$
```



**Figure 1**



**Figure 2**